BLACK CODHER

CODING PROGRAMME

Black Codher Bootcamp

BLACK CODHER

CODING PROGRAMME

UNIT 4 - Session 7
React

Session 6 Summary

BLACK CODHER
CODING PROGRAMME

1. React routes
2. React web pages
3. Fragments
4. Form Management
5. Controlled Components
6. Continued with the Bookcase App

3

Goals for Unit 4 - Session 7

BLACK CODHER
CODING PROGRAMME

1. What is an API?
2. Public APIs
3. RESTful APIs
4. Google Book API and how to connect to it
5. Fetch API
6. React Bootstrap

4

BLACK CODHER

CODING PROGRAMME

What is an API?

What is an API?

BLACK CODHER
CODING PROGRAMME

- API stands for Application Programming Interface
- An API is software that sits in-between two applications and allows them to talk to each other
- APIs allow us to share important data and expose practical business functionality between devices, applications, and individuals
- One of the chief advantages of APIs is that they allow the abstraction of functionality between one system and another
- APIs enable developers to make repetitive yet complex processes highly reusable with a little bit of code

Public APIs

Why is it helpful to know what an API is and how to consume one?

- Most public application have APIs that can be consumed by external programs nowadays

- Facebook, Twitter, Instagram, Meetup, Eventbrite and many more will allow you to search or request data from their system which can enhance any application you are building

Why would they allow this?

- Allow this extends the reach of an application by allowing third-party developers to easily and safely build extended programs

Public APIs

BLACK CODHER
CODING PROGRAMME

- There are a number of open APIs available for developers to query
- Most APIs will require you to sign up and get an API key to use
- Some interesting Public APIs:

https://github.com/public-apis/public-apis

- The link contains a list of over 100 public APIs that can be used in your projects. It contains APIs that return data on anything from animals, food, government and weather
- The majority of APIs available will be **RESTful** APIs though other architectures exists

What are RESTful APIs?

BLACK CODHER
CODING PROGRAMME

- REST or RESTful API (stands for **RE**presentational **S**tate **T**ransfer)
- REST is designed to take advantage of existing web protocols
- REST Services are generally stateless, meaning the server does not store any data about the client session on the server
- The client session information is stored by the client
- Other API architectures exist such as SOAP (**S**imple **O**bject **A**ccess **P**rotocol)
- We'll cover the specifics of APIs in more detail in the Node.js course
- Next we will be looking specifically at the RESTful Google Book API

BLACK CODHER
CODING PROGRAMME

Google Book API

10

Google Book API

BLACK CODHER
CODING PROGRAMME

- The Google Books API is Google's effort to make book content more discoverable on the Web
- Using the Google Books API, your application can perform full-text searches and retrieve book information, viewability and eBook availability
- You can also manage your own personal bookshelves
- Most calls to the Google API require authentication using an API key
- Performing a search does not require authentication, so you do not have to provide the Authorization header with a request to the API
- We'll be using the Fetch API to make request to the Google Book API

11

Google Book API

- The Google Books API returns JSON in the same format as the books.json file. Entering the following URL into a browser window will result in a list of books about food:
  https://www.googleapis.com/books/v1/volumes?q=food&filter=paid-ebooks&print-type=books&projection=lite


```
{
  "kind": "books#volumes",
  "totalItems": 1730,
  "items": [
    {
      "kind": "books#volume",
      "id": "_t0IoTcVxIIC",
      "etag": "M+sHQ1WlR64",
      "selfLink": "https://www.googleapis.com/books/v1/volumes/_t0IoTcVxIIC",
      "volumeInfo": {
        "title": "The Psychology of Food Choice",
        "authors": [
          "Richard Shepherd","Monique Raats"],
        "description": "One of the central problems in nutrition is ...",
       ...
```


12

BLACK CODHER

CODING PROGRAMME

Exercise 1

13

Exercise 1: Adding a Search Form

BLACK CODHER
CODING PROGRAMME

- We are going to add search capabilities to our bookcase app. First we will need to add a search form to the existing app

1. Open the **mybookcase** app

2. Create a new component file called **Search.js** and place it in the components folder

```
> cd mybookcase/src
> echo . > Search.js
```


14

Exercise 1: Adding a Search Form

3. Open **Search.js** in Visual Code. Add a React functional component called Search

```
import React from 'react';
const Search = (props) => {
  return <div> ...Add input/submit button here </div>
}
export default Search;
```

4. Return a form from the Search component. The form should contain an input field to collect the search term entered by a user and a submit button

5. Add a reference to your Search component in your **Header.js** file e.g.

15

Exercise 1: Adding a Search Form

```
<Route exact path="/" render={() => (
    <>
      <Header/>
      <Search />
      <BookList />
    </>
)} />
```

6. Run the code from the terminal to ensure the form is returned (e.g. `> npm start`)

7. Ensure the input field is a controlled component by setting the value of the input field to a props attribute passed in by the calling component e.g.

16

Exercise 1: Adding a Search Form

```
<input type="text" value={keyword} onChange={(e)
=>setKeyword(e.target.value)}/>
```

8. You will need to call a function called **setKeyword** which is returned by the **useState** hook e.g.

```
const [keyword,setKeyword] = useState('');
```

9. To test if the **setKeyword**() function is working, you can display the results of the keyword somewhere on the form e.g.

```
<form>
  <h1>{keyword && 'Searching for keyword:' + keyword}</h1>
...The rest of your form fields e.g.<input/> <button/>
</form>
```


17

Checkpoint!

BLACK CODHER
CODING PROGRAMME

How are you feeling?

RED - I have no idea what you're talking about.

YELLOW - I have some questions but feel like I understand some things.

GREEN - I feel comfortable with everything you've said.

BLACK CODHER
CODING PROGRAMME

Using Fetch in React

19

Fetch in React

BLACK CODHER
CODING PROGRAMME

- Consuming REST APIs in a React Application can be done in various ways
- We'll be looking at the Fetch API which is an in-built browser web API that provides an interface for fetching resources
- A fetch function is now provided in the global window scope, with the first argument being the URL.

Syntax: `fetch(<some API endpoint URL>, {method: <get|post|put|delete>});`

e.g. `fetch('https://www.example.com/some/url', {method: 'get'});`

- We'll cover request methods in detail in the next course. There a few different methods you can use - we will mainly be using the GET method

20

Fetch in React

- The call below is an example of using the fetch API to search the Google API for books about 'food':

```
const term = 'food';
const results = await
fetch(`https://www.googleapis.com/books/v1/volumes?q=${term}&filter=paid-ebooks&print-type=books&projection=lite`).then(res => res.json());
```

- The GET method is the default so doesn't have to be specified
- The term food is being interpolated into the URL
- The call above also uses Javascript promises to handle requests/callbacks.

21

Fetch in React

BLACK CODHER
CODING PROGRAMME

- The response is a JSON list of books

- To call the Google API from our Bookcase app we can add a function called **`findBooks()`** in our App.js file which will update a state variable called **books** via a **`setBooks()`** function

```
import data from './models/books.json';
const [books, setBooks] = useState(data);

async function findBooks(value) {
    const results = await
fetch(`https://www.googleapis.com/books/v1/volumes?q=${value}&filter=paid-ebooks&print-ty
pe=books&projection=lite`).then(res => res.json());
   if (!results.error) {
     setBooks(results.items);
   }
}
```


22

Fetch in React

BLACK CODHER
CODING PROGRAMME

- The promises/async/await pattern is a syntax that enables asynchronous, non-blocking functions to be structured in a way similar to synchronized functions

- Synchronous code is executed in sequence – each statement waits for the previous statement to finish before executing.

- Asynchronous code doesn't have to wait – your program can continue to run. You do this to keep your site or app responsive, reducing waiting time for the user

23

BLACK CODHER

CODING PROGRAMME

Exercise 2

24

Exercise 2: Search Google Book API

BLACK CODHER
CODING PROGRAMME

- The search will take a text input from a user and search the Google Books API for a list of books that match the term

- The results of the search will be displayed to the user. Each book will have an "Add" button to enable a user to add the book to a personal list of books

1. Open the **mybookcase** app. Open the App.js file in Visual Code

2. Add a function called **findBooks()** which calls the Google Book API using the fetch API

3. In your **findBooks**() function in the **App.js** ensure it takes a variable which will be set/sent from the Search component

25

Exercise 2: Search Google Book API

4. You will need to pass the function **findBooks** down to the Search component via its attributes

```
<Search findBooks={findBooks}/>
```

- Google Books API endpoint:

```
const value = 'food';
https://www.googleapis.com/books/v1/volumes?q=${value}&filter=paid-ebooks
&print-type=books&projection=lite
```

- This endpoint currently filters on paid-ebooks, print-type: books and projection: lite

- For more details of the API visit the homepage: https://developers.google.com/books/docs/v1/reference/

26

Exercise 1: Search the Google API

BLACK CODHER
CODING PROGRAMME

5. In your Search.js file add a method to handle the form submit, e.g. onSubmit={handleSubmit}

6. In the handleSubmit method remember to prevent the form form posting a request by calling event.preventDefault()

```
<form onSubmit={handleSubmit}>...</form>

const handleSubmit = (event) =>{
  event.preventDefault();
  props.findBooks(props.keyword);
};
```


27

Exercise 1: Search the Google API

BLACK CODHER
CODING PROGRAMME

7. You will need to lift the state of the **keyword** field up to the App.js and pass it back down to the Search component via the attributes e.g

```
<Search findBooks={findBooks} keyword={keyword} setKeyword={setKeyword}/>
```

8. Update the form elements in Search.js and set the text field's value attribute to **props.keyword** and call the **props.setkeyword()** from the onChange e.g.

```
<input value={props.keyword} onChange={(e) =>
props.setKeyword(e.target.value)}/>
```

9. Ensure the books list is updated with the return value from the Google Books API

10. Test your app by running the npm start command

28

BLACK CODHER
CODING PROGRAMME

React Bootstrap

29

React Bootstrap

- There are a number of React libraries that you can incorporate into your applications to help improve the look and feel or enhance the functionality of your site

- Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development

- It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components

- The React-Bootstrap library will give you access to a number of pre-built components as well as css classes for styling your applications

- For a comprehensive list of components visit the react-bootstrap component library page: https://react-bootstrap.github.io/components

30

React Bootstrap

- NPM can be used to add the React Bootstrap package to your React application

```
> npm install react-bootstrap bootstrap
```

- Once installed you can reference the React-bootstrap css from your App.js file

```
import 'bootstrap/dist/css/bootstrap.min.css';
```


31

React Bootstrap

BLACK CODHER
CODING PROGRAMME

- You can also import individual components like react-bootstrap/Button rather than the entire library

```
import Button from 'react-bootstrap/Button';

import { Button } from 'react-bootstrap';
```

- Doing so pulls in only the specific components that you use, which can significantly reduce the amount of code you end up sending to the clien

32

React Bootstrap

- Reference your imported component in your functional component e.g.

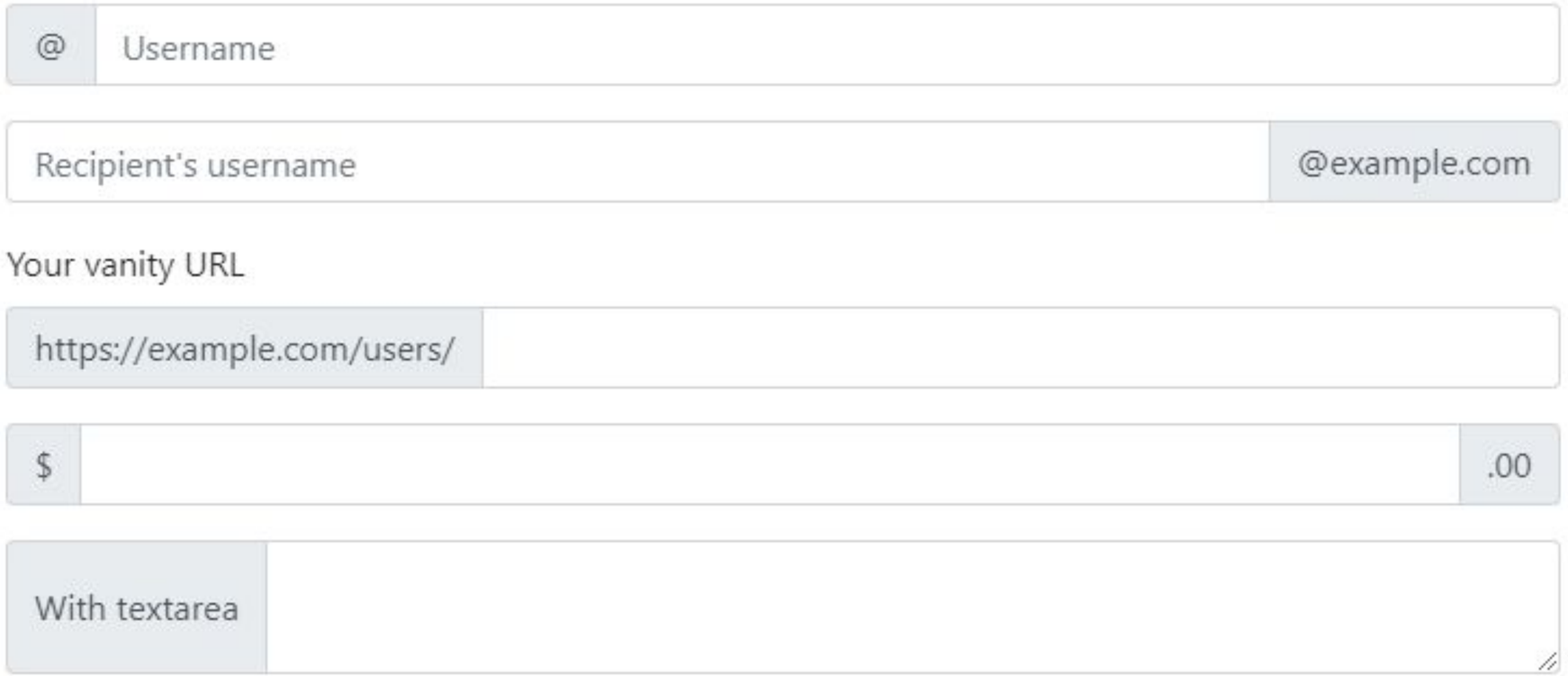
```
const Example = ({ props }) => {

  return  <>
          <Button className="btn-default">Default Button</Button>
          <Button className="btn-warning">Warning Button</Button>
          <Button className="btn-info">Info Button</Button>
          <Button className="btn-danger">Danger Button</Button>
          <Button className="btn-success">Sucess Button</Button>
          <Button className="btn-dark">Dark Button</Button>
          <Button className="btn-light">Light Button</Button></>
}
export default Example;
```

33

React Bootstrap

- There are a number of classes and components that can be added to improve the usability of forms and input fields: e.g.



34

BLACK CODHER

CODING PROGRAMME

Exercise 1

35

Exercise 3: Adding Bootstrap

BLACK CODHER
CODING PROGRAMME

- We are going to add react-bootstrap to our library to improve the look-and-feel of the app

1. Open the **mybookcase** app

2. Navigate to the application from the terminal/command line and run the install command

```
> npm install react-bootstrap bootstrap
```

3. Add a reference to the bootstrap css library to the App.js file e.g.

```
import 'bootstrap/dist/css/bootstrap.min.css';
```


36

Exercise 3: Adding Bootstrap

BLACK CODHER
CODING PROGRAMME

4. Update your form elements. Change them to bootstrap input fields

```
<Form>
  <Form.Group controlId="searchKeyword">
    <Form.Label>Enter Search</Form.Label>
    <Form.Control type="keyword" placeholder="Enter keyword" />
  </Form.Group>

  <Button variant="primary" type="submit">
    Submit
  </Button>
</Form>
```

5. Add the findBooks(), HandleSubmit() and keyword attributes and events to ensure your Bootstrap form works with your exisiting code

37

Checkpoint!

BLACK CODHER
CODING PROGRAMME

How are you feeling?

RED - I have no idea what you're talking about.

YELLOW - I have some questions but feel like I understand some things.

GREEN - I feel comfortable with everything you've said.

Summary of Session 7

BLACK CODHER
CODING PROGRAMME

1. What is an API?
2. RESTful APIs
3. Google Book API and how to connect to it
4. Fetch API
5. React Bootstrap

39